

*Citation for published version:*

Sun, L & Zang, J 2013, 'OpenMP implementation for Fortran on HPC', HPC symposium , Bath, UK United Kingdom, 4/06/13 - 4/06/13.

*Publication date:*

2013

*Document Version*

Early version, also known as pre-print

[Link to publication](#)

## University of Bath

### Alternative formats

If you require this document in an alternative format, please contact:  
[openaccess@bath.ac.uk](mailto:openaccess@bath.ac.uk)

#### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

#### Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

# OpenMP implementation for FORTRAN on HPC

Speaker: Dr **Liang Sun**

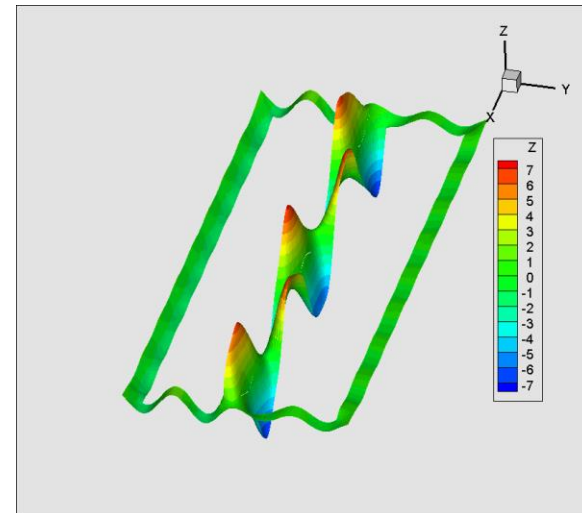
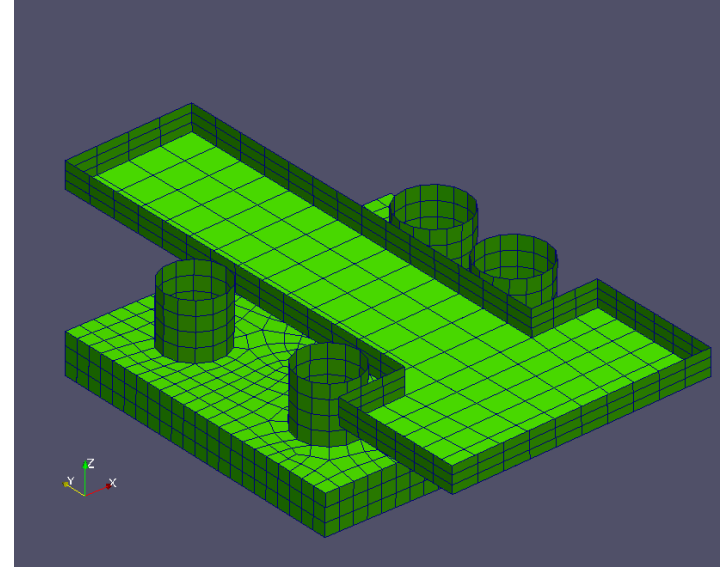
Email: [l.sun@bath.ac.uk](mailto:l.sun@bath.ac.uk)

Co-author: Dr **Jun Zang**

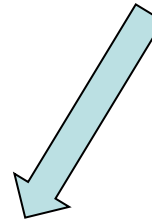
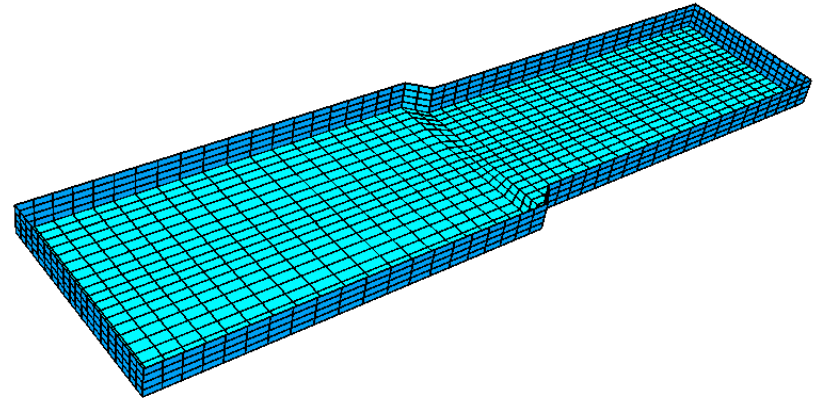
Email: [j.zang@bath.ac.uk](mailto:j.zang@bath.ac.uk)

*Department of Architecture and Civil Engineering  
University of Bath*

# Background

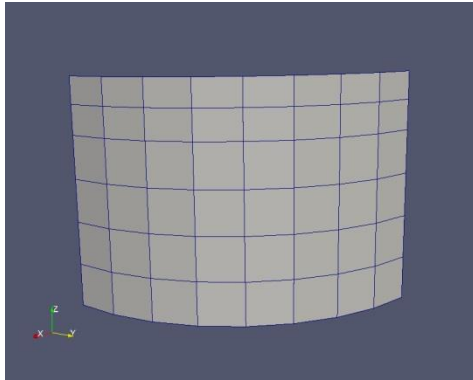


# Numerical method

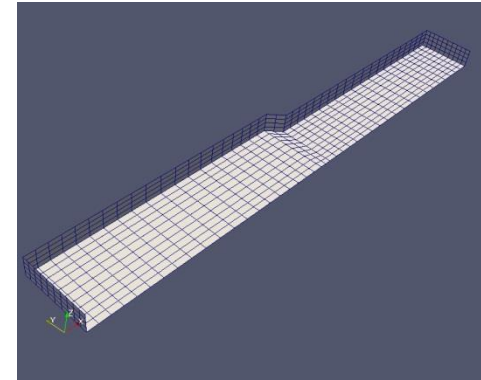
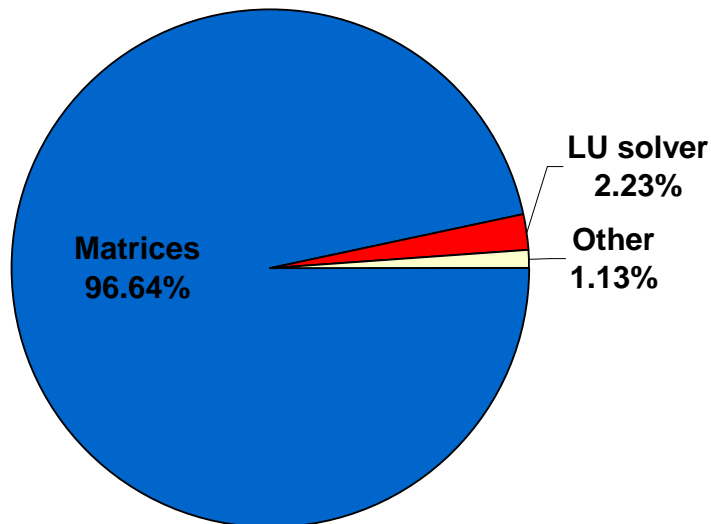


$$[A]\{x\} = \{b\}$$

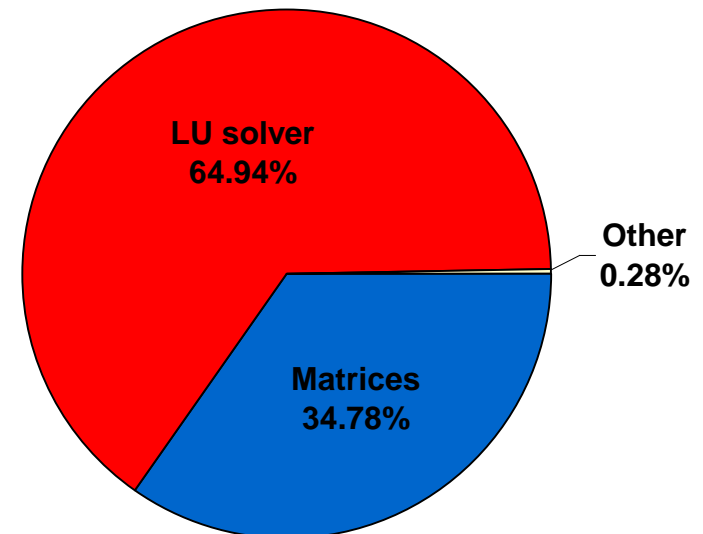
# Performance of sequential executable



$$[A]_{264 \times 264}$$



$$[A]_{3952 \times 3952}$$



# Better LU solver in Intel MKL

## LAPACK Routines: Linear Equations

### ?gesv

*Computes the solution to the system of linear equations with a square matrix A and multiple right-hand sides.*

---

### Syntax

#### Fortran 77:

```
call zgesv( n, nrhs, a, lda, ipiv, b, ldb, info )
```

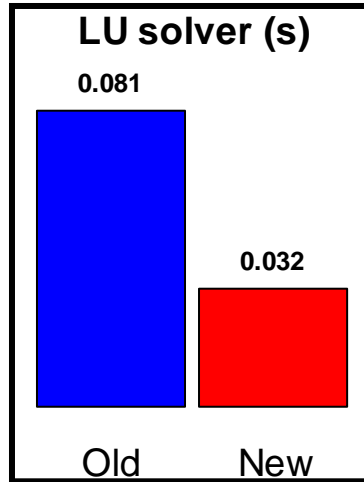
#### Reference:

Intel® Math Kernel Library Reference Manual

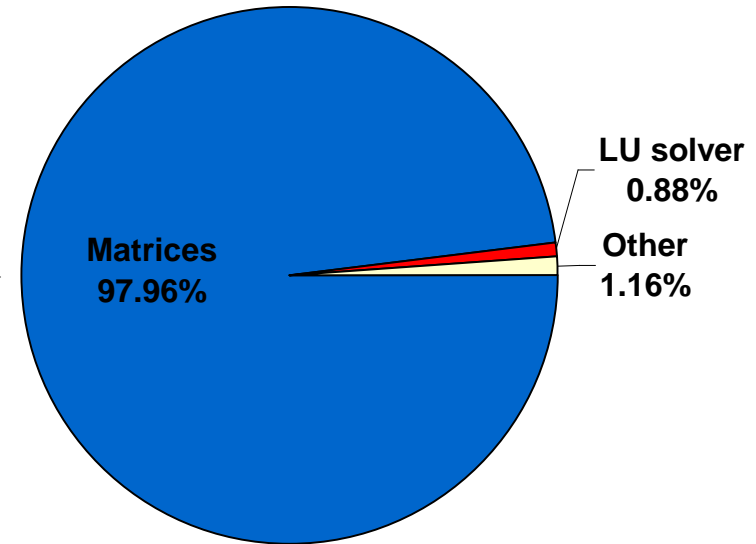
<http://software.intel.com/en-us/articles/intel-math-kernel-library-documentation>

# Performance of sequential executable with MKL

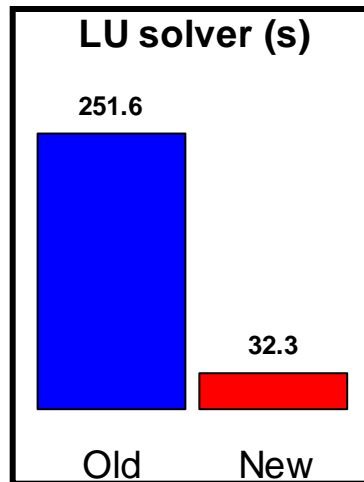
$[A]_{264 \times 264}$



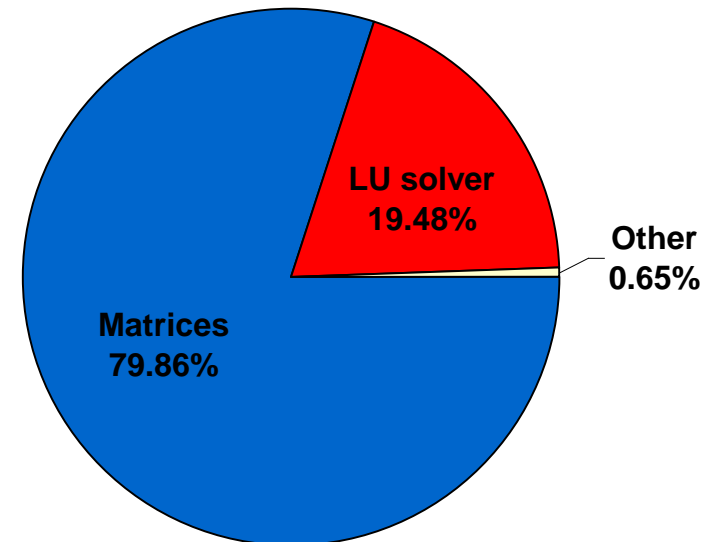
save **60%**



$[A]_{3952 \times 3952}$

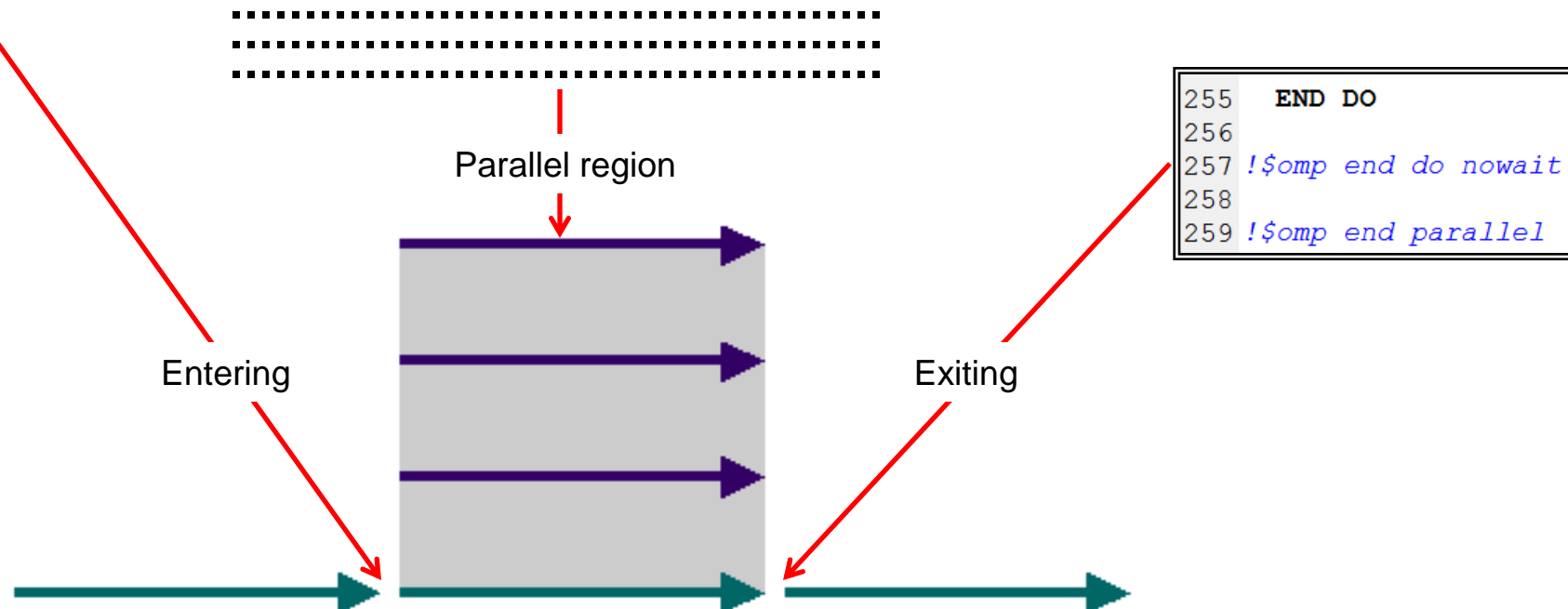


save **87%**



# Implementation of OpenMP

```
42 !$omp parallel default(none)  &  
43 !$omp shared(node_body,xyz_p,xyz,amata,ncon,ncon_p,rsn,nphi,nsys,v,nele_body,ncn,bmata,nnode_p,nelem) &  
44 !$omp private(inode,xp,yp,zp,value,bmat,ielem,i,check,wmat,wmatl,ip,j,jncon,ith,is,xyzco,el,dist,kk)  
45  
46 !$omp do  
47 ! FOR EQUATIONS ON THE BODY SURFACE  
48  
49 DO INODE=1,NODE_BODY           !SOURECE POINTS ON BODY SURFACE
```





# Data race



## ➤ Data-Sharing Attributes

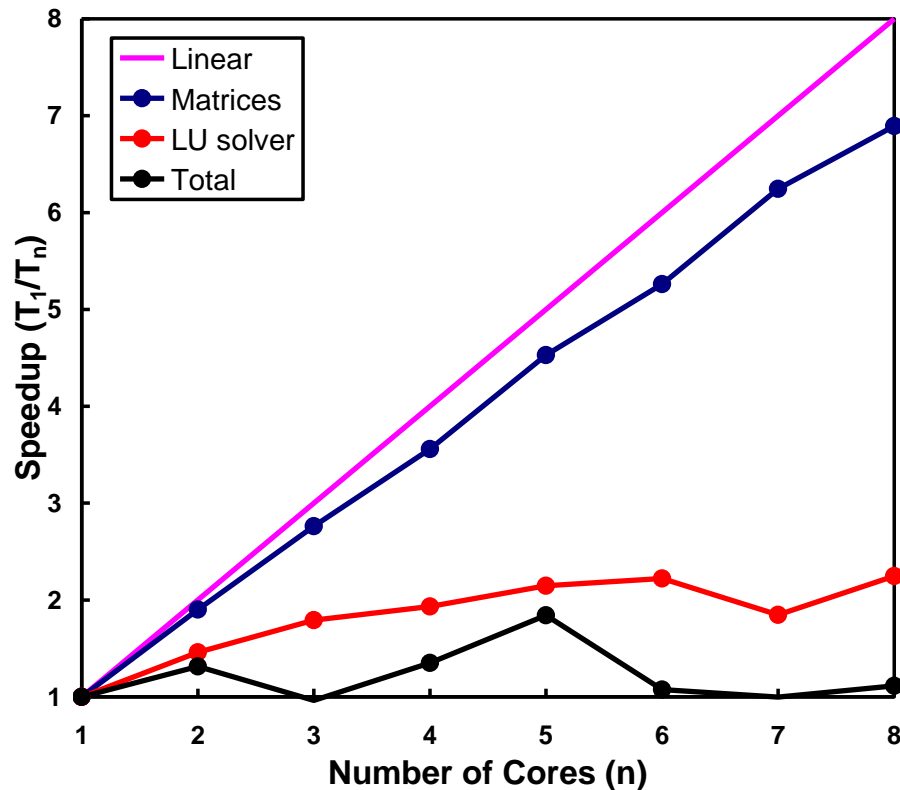
```
42 !$omp parallel default(none) &  
43 !$omp shared(node_body,xyz_p,xyz,amata,ncon,ncon_p,rsn,nphi,nsys,v,nele_body,ncn,bmata,nnode_p,nelem) &  
44 !$omp private(inode,xp,yp,zp,value,bmat,ielem,i,check,wmat,wmatl,ip,j,jncon,ith,is,xyzco,el,dist,kk)
```

## ➤ Threadprivate Directive

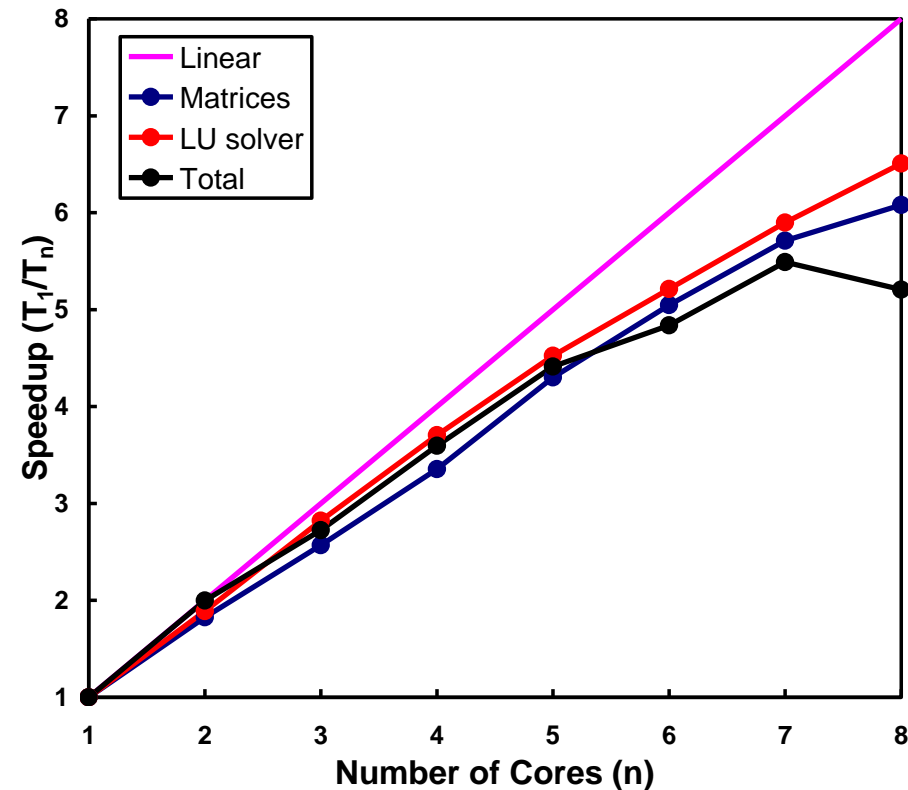
```
107 C$omp threadprivate(/FGRIGR/,/HCOEF/)
```

# Speedup of parallel executable

$$[A]_{264 \times 264}$$



$$[A]_{3952 \times 3952}$$

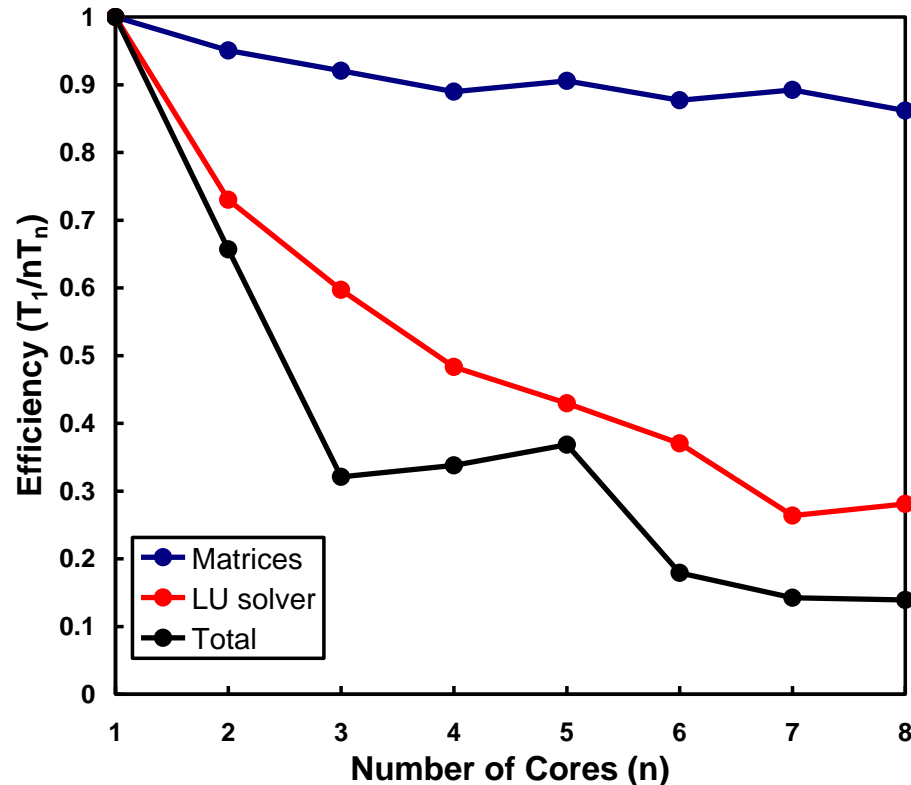


$T_1$  is the execution time of the sequential algorithm

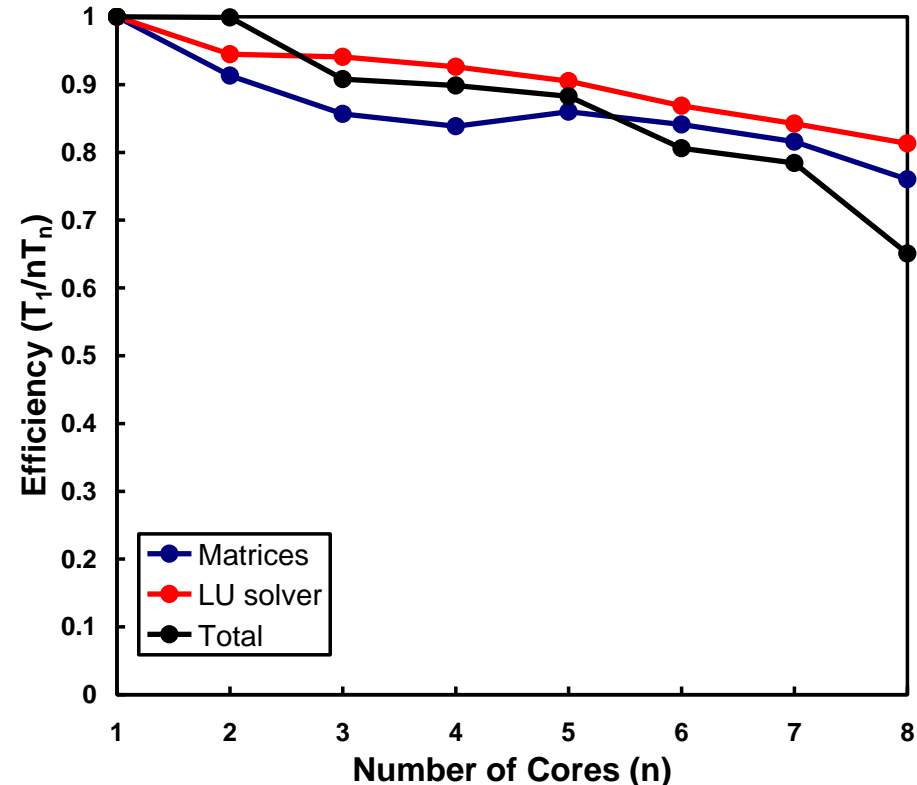
$T_n$  is the execution time of the parallel algorithm with  $n$  cores

# Efficiency of parallel executable

$[A]_{264 \times 264}$



$[A]_{3952 \times 3952}$



$T_1$  is the execution time of the sequential algorithm

$T_n$  is the execution time of the parallel algorithm with  $n$  cores

# Concluding remarks

- Optimized LU solver in Intel MKL improves performance significantly
- OpenMP has been implemented successfully in current FORTRAN codes and all data race problems have been solved
- Running multithreaded executable for small problems is not economical considering total computational time. In large problems, much time can be saved by using parallel algorithm

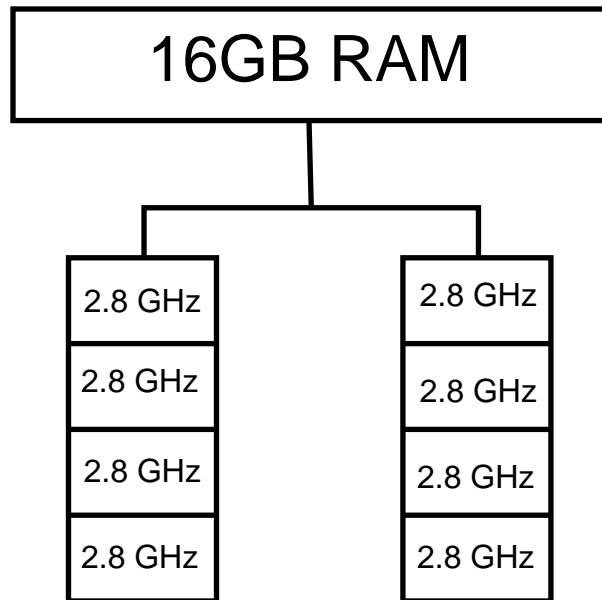
# Acknowledgement

These computations were performed on the University of Bath's High Performance Computing Facility. Provision of services by BUCS HPC Support Team is gratefully acknowledged.

# **Thank You !**

# **Additional Information**

# Hardware and software on HPC



HPC Node

Intel FORTRAN Compiler  
Module: *icomp/11.1.075*

Intel Math Kernel Library (MKL)  
Module: *imkl/10.2.7.041*

Compiler and Library



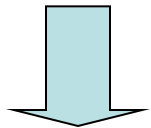
# Generation of sequential executable

```
GNU Make 3.81
Copyright (C) 2006 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.
There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A
PARTICULAR PURPOSE.

This program built for x86_64-redhat-linux-gnu
```



Code::Blocks



Makefile

```
SRC_DIR_f90d1 = /home/ls650/codes/D_org/src/
```

```
SRC_DIR_fd1 = /home/ls650/codes/D_org/src/
```

```
OBJS_DIR = /home/ls650/codes/D_org/obj/
```

```
EXE_DIR = /home/ls650/codes/D_org/bin/
```

```
EXE = D_org
```

```
FC = ifort
```

```
IDIR =
```

```
CFLAGS = -fast -module $(OBJS_DIR) $(IDIR)
```

```
LFLAGS = -s
```

```
LIBS =
```

# Generation of sequential executable with MKL

Intel® Math Kernel Library (MKL) Link Line Advisor v2.2 Reset

Select Intel® product:	Intel(R) MKL 10.2
Select OS:	Linux*
Select compiler:	Intel(R) Fortran
Select architecture:	Intel(R) 64
Select dynamic or static linking:	Dynamic
Select interface layer:	LP64 (32-bit integer)
Select sequential or multi-threaded layer:	Sequential
Select OpenMP library:	<Select OpenMP>
Select cluster library:	<input type="checkbox"/> CDFT (BLACS required) <input checked="" type="checkbox"/> ScaLAPACK (BLACS required) <input type="checkbox"/> BLACS
Select MPI library:	<Select MPI>
Select the Fortran 95 interfaces:	<input type="checkbox"/> BLAS95 <input checked="" type="checkbox"/> LAPACK95
Link with Intel® MKL libraries explicitly:	<input type="checkbox"/>

Use this link line:

```
-L$(MKLROOT)/lib/em64t $(MKLROOT)/lib/em64t/libmkl_lapack95_lp64.a -lmkl_intel_lp64 -lmkl_sequential -lmkl_core -lpthread -lm
```

Compiler options:

```
-I$(MKLROOT)/include/em64t/lp64 -I$(MKLROOT)/include
```

```
CFLAGS = -fasm -module $(OBJDIR) $(IDIR)  
LFLAGS =  
LIBS = -L$(MKLRROOT)/lib/em64t $(MKLRROOT)/lib/em64t/libmkl lapack95 lp64.a -lmkl intel_lp64 -lmkl sequential -lmkl core -lpthread -lm
```

# Generation of parallel executable

### Intel® Math Kernel Library (MKL) Link Line Advisor v2.2

Reset

Select Intel® product:	Intel(R) MKL 10.2
Select OS:	Linux*
Select compiler:	Intel(R) Fortran
Select architecture:	Intel(R) 64
Select dynamic or static linking:	Dynamic
Select interface layer:	LP64 (32-bit integer)
Select sequential or multi-threaded layer:	Multi-threaded
Select OpenMP library:	Intel(R) (libiomp5)
Select cluster library:	<input type="checkbox"/> CDFT (BLACS required) <input type="checkbox"/> ScaLAPACK (BLACS required) <input type="checkbox"/> BLACS
Select MPI library:	<Select MPI>
Select the Fortran 95 interfaces:	<input type="checkbox"/> BLAS95 <input checked="" type="checkbox"/> LAPACK95
Link with Intel® MKL libraries explicitly:	<input type="checkbox"/>

Use this link line:

```
-I$(MKLROOT)/lib/em64t $(MKLROOT)/lib/em64t/libmkl_lapack95_lp64.a -lmkl_intel_lp64 -lmkl_intel_thread -lmkl_core -lpthread -lm
```

Compiler options:

```
-openmp -I$(MKLROOT)/include/em64t/lp64 -I$(MKLROOT)/include
```

```
CFLAGS = -fast -openmp ██████████ -module $(OBJDIR) $(IDIR)  
LFLAGS = -s  
LIBS = -I$(MKLRROOT)/lib/em64t $(MKLRROOT)/lib/em64t/libmkl_lapack95_lp64.a -lmkl_intel_lp64 -lmkl_intel_thread -lmkl_core -lpthread -lm -liomp5
```

# References

- <https://wiki.bath.ac.uk/display/HPC/OpenMP>
- Intel® Math Kernel Library Reference Manual (<http://software.intel.com/en-us/articles/intel-math-kernel-library-documentation>)
- Intel® Math Kernel Library Link Line Advisor (<http://software.intel.com/en-us/articles/intel-mkl-link-line-advisor>)
- Chapman, B., Jost, G., van der Pas, R., 2007. [Using OpenMP: Portable Shared Memory Parallel Programming](#). MIT Press, Cambridge, Massachusetts, USA.
- [www.openmp.org](http://www.openmp.org)